I faced two major obstacles when coding this project. The first was with the hasCorrectSyntax function. I originally had if statements of the form *if (beat.size() – 2 >= 0)*, but this always returned true. I am still not entirely sure as to why, but this problem was fixed by rewriting the if statement into a form of *if (beat.size() >= 2)*. The second major obstacle was with converting freezes and the associated errors. My mistakes in that section of code were almost entirely off-by-one errors, and carefully examining what my code was doing step by step allowed me to fix the code with relative ease.

My hasCorrectSyntax function first checks that only the allowed characters are in the given string before checking each beat for having correct syntax:

*repeatedly (for dance length):*

*check that character is allowed*

*if end of beat is found*

*find start of next beat*

*if beat instructions do not have proper syntax*

*return false*

*if there are leftover characters after final beat*

*return false*

*return true (as default)*

My convertDance function first checks if there is correct syntax by accessing the hasCorrectSyntax function and will convert the dance unless an error is run into, in which case it will return the corresponding error number and provide the location of the error.

*if there is incorrect syntax*

*return 1*

*repeatedly (for dance length):*

*if end of beat is found*

*find start of next beat*

*if beat indicates a freeze*

*calculate freeze length*

*if freeze length is 0 or 1*

*increment beat number*

*bad beat becomes beat number*

*return 3*

*repeatedly (for freeze length):*

*increment beat number*

*if the end of the dance has been reached*

*bad beat becomes beat number*

*return 2*

*if some other instruction appears during the freeze*

*bad beat becomes beat number*

*return 4*

*add a capital direction letter to the converted dance*

*adjust start of beat to account for freeze*

*if beat indicates nothing or a tap*

*add appropriate character to converted dance*

*assign the converted dance to the indicated variable*

*return 0 (as default)*

Test data:

Correct syntax and convertible:

Empty string (“”)

1 beat, no direction (“/”)

Tap, lowercase (“r/”)

Tap, uppercase (“L/”)

Freeze (1 digit), lowercase (“2u//”)

Freeze (1), uppercase (“2D//”)

Freeze (2), lowercase (“03l///”)

Freeze (2), uppercase (“03R///”)

Combination of all of the above (“/03D////U/d/2L//2r//03u///”)

Incorrect syntax:

Disallowed characters (“/r/T/l/2D//”)

Characters after last beat (“r/L/3u///r”)

Disallowed beat format, just a number (“2/”)

Disallowed beat format, letter then letter (“rr/”)

Disallowed beat format, letter then number (“u2/”)

Disallowed beat format, letter, number, then letter (“r2l/”)

Disallowed beat format, 3 numbers (“013r/”)

Correct syntax but inconvertible:

A beat not consisting of only a slash during a freeze (“3r/u/d/”)

Ends prematurely (“5r//”)

Freeze length of 0 (“u//0r/”)

Freeze length of 1 (“3r///1U//”)

Multiple errors at different locations (“3r//u/0d//2u/”)

Multiple errors at the same location (“3r//0d/”)

Convertible freeze followed by inconvertible (prematurely ending) freeze (“03d///4l//”)

Convertible freeze followed by inconvertible (non-slash beat) freeze (“4U////3d/r//”)